IN THE CLAIMS

Please cancel Claims 1 – 32 and 35 without prejudice, and add new Claims 36 – 61 as follows:

1. – 32. (Cancelled)

33. (Previously presented) A method of debugging a plurality of processes on a plurality of heterogeneous processors using a debugger program, comprising:

initializing a poll delay associated with each process;

determining the state of said debugger program;

if said state comprises a first value, determining the run status of at least one of said processes;

if said at least process is running, determining a process type associated with said at least one process;

if said process type comprises a simulation type, advancing through at least one instruction cycle and checking the status of said simulated process; and

if said process type comprises a hardware type, delaying said debugger program for a predetermined time until a subsequent polling opportunity is available, and determining the status of said hardware process at or after said subsequent polling opportunity;

wherein said plurality of processes are adapted to gather status information regarding respective ones of said heterogeneous processors as controlled by said debug process, said gathering of status information occurring on a dynamic per-process time interval.

34. (Previously presented) The method of Claim 33, further comprising:

utilizing said debugger program to check a value of simulator variable to determine whether any currently running ones of said processes are executing in a simulator; and if so, returning said debugger program immediately to said act of determining the state so that said process(es) executing in a simulator will run as quickly as possible.

35. (Cancelled)

36. (New) The method of Claim 33, further comprising analyzing said status information to identify at least one or more occurrences of errors.

37. (New) The method of Claim 33, further comprising:

defining at least one object class;

defining at least one first object subclass for said hardware type process; and

defining at least one second object subclass for said simulation type process.

38. (New) The method of Claim 33, further comprising dynamically changing polling times associated with at least one of said plurality of processes based on the status thereof.

39. (New) The method of Claim 33, further comprising defining an interface to a first library for said hardware type process.

40. (New) The method of Claim 39, further comprising accessing said first library via said interface in order to provide functions relating to at least one extension instruction.

41. (New) The method of Claim 33, further comprising defining an interface to a first library for said simulation type process.

42. (New) The method of Claim 41, further comprising accessing said first library via said interface in order to provide functions relating to at least one extension instruction, including the implementation of said at least one extension instruction.

43. (New) The method of Claim 33, further comprising:

defining a plurality of individual subclasses for each of said plurality of processes; and

implementing at least a portion of said subclasses as a dynamically loadable library.

44. (New) The method of Claim 33, wherein said gathering of status information comprises switching without determining whether or not a debug event has occurred.

45. (New) The method of Claim 33, wherein said act of gathering status information comprises obtaining said status information without determining whether a debug event has occurred.

46. (New) An apparatus for debugging a plurality of processes on a plurality of heterogeneous processors, said apparatus debugging according to the method comprising:

initializing a poll delay associated with each process;

determining the state of said debugger program;

if said state comprises a first value, determining the run status of at least one of said processes;

if said at least process is running, determining a process type associated with said at least one process;

if said process type comprises a simulation type, advancing through at least one instruction cycle and checking the status of said simulated process; and

if said process type comprises a hardware type, delaying said debugger program for a predetermined time until a subsequent polling opportunity is available, and determining the status of said hardware process at or after said subsequent polling opportunity;

wherein said plurality of processes are adapted to gather status information regarding respective ones of said heterogeneous processors as controlled by said debug process, said gathering of status information occurring on a dynamic per-process time interval.

47. (New) The apparatus of Claim 46, wherein said plurality of heterogeneous processors comprises a digital processor embodied as an integrated circuit, and said debugging method comprises is perfromed by a computer program adapted to run on said integrated circuit.

48. (New) The apparatus of Claim 46, further comprising at least one external port adapted for data communication with respective ones of said hardware type processes.

49. (New) A method of debugging a plurality of processes on a plurality of heterogeneous processors using a debugger program, comprising:

initializing a poll delay associated with each process;

determining the state of said debugger program;

determining the run status of at least one of said processes; and

determining a process type associated with said at least one process, wherein if said process type comprises a simulation type, advancing through at least one instruction cycle and checking the status of said simulated process; and

wherein if said process type comprises a hardware type, delaying said debugger program for a predetermined time until a subsequent polling opportunity is available, and determining the status of said hardware process at or after said subsequent polling opportunity; and

wherein said plurality of processes are adapted to gather status information regarding respective ones of said heterogeneous processors as controlled by said debug process, said gathering of status information occurring on a dynamic per-process time interval.

50. (New) The method of Claim 49, further comprising:

utilizing said debugger program to check a value of simulator variable to determine whether any currently running ones of said processes are executing in a simulator; and if so, returning said debugger program immediately to said act of determining the state.

51. (New) The method of Claim 49, further comprising analyzing said status information to identify at least one or more occurrences of errors.

52. (New) The method of Claim 49, further comprising:

defining at least one object class,

defining at least one first object subclass for said hardware type process; and

defining at least one second object subclass for said simulation type process.

53. (New) The method of Claim 49, further comprising dynamically changing polling times associated with at least one of said plurality of processes based on the status thereof.

54. (New) The method of Claim 49, further comprising defining an interface to a first library for said hardware type process.

55. (New) The method of Claim 54, further comprising accessing said first library via said interface in order to provide functions relating to at least one extension instruction.

56. (New) The method of Claim 49, further comprising defining an interface to a first library for said simulation type process.

57. (New) The method of Claim 56, further comprising accessing said first library via said interface in order to provide functions relating to at least one extension instruction, including the implementation of said at least one extension instruction.

58. (New) The method of Claim 49, further comprising:

defining a plurality of individual subclasses for each of said plurality of processes; and

implementing at least a portion of said subclasses as a dynamically loadable library.

59. (New) The method of Claim 49, wherein said gathering of status information comprises switching without determining whether or not a debug event has occurred.

60. (New) The method of Claim 49, wherein said act of gathering status information comprises obtaining said status information without determining whether a debug event has occurred.

61. (New) A method of debugging a plurality of processes on a plurality of heterogeneous processors using a debugger program, comprising:

a step for initializing a minimum time period between retrievals of said heterogeneous processor status information for display to a user with each process;

a step for determining the state of said debugger program;

a step for determining the run status of at least one of said processes if said state comprises a first value;

a step for determining a process type associated with said at least one process if said at least process is running;

a step for advancing through at least one instruction cycle and checking the status of said simulated process if said process type comprises a simulation type; and

a step for delaying said debugger program for a predetermined time until a subsequent polling opportunity is available if said process type comprises a hardware type; and

a step for determining the status of said hardware process at or after said subsequent polling opportunity;

wherein said plurality of processes are adapted to gather status information regarding respective ones of said heterogeneous processors as controlled by said debug process, said gathering of status information occurring on a dynamic per-process time interval.